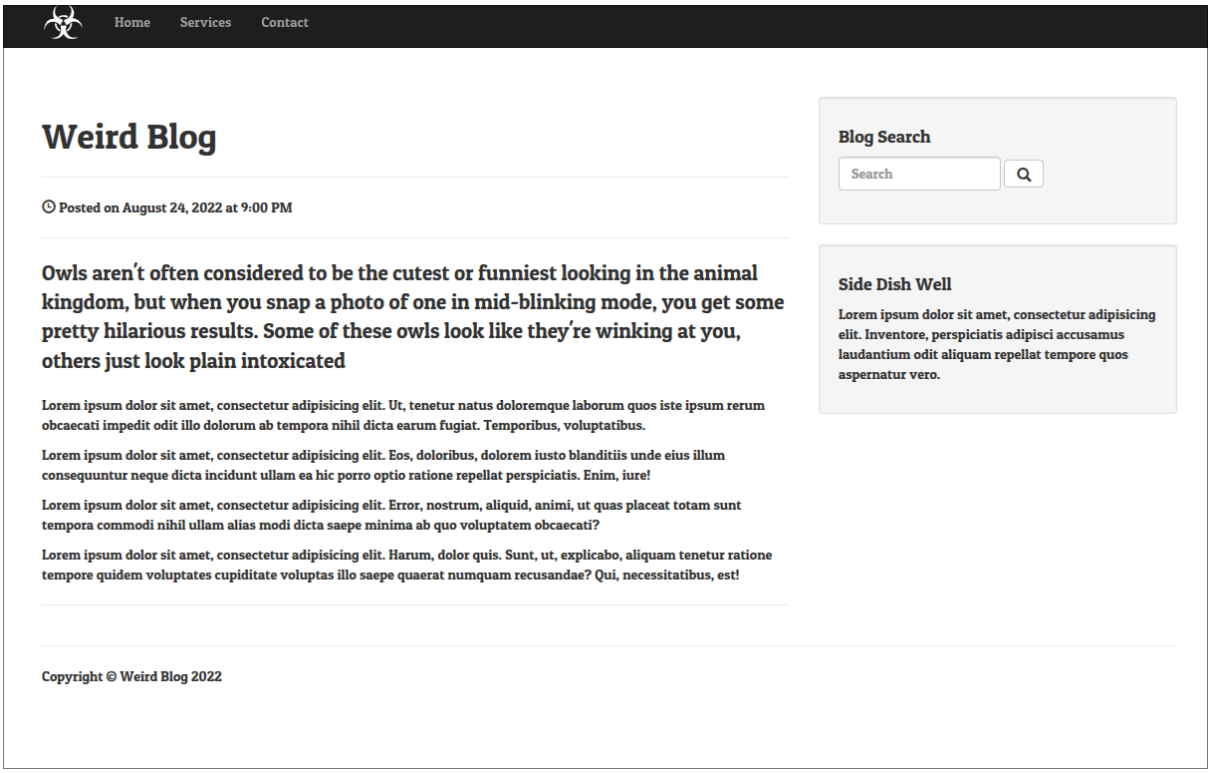
CyberTalents Challenges

Web Security – Owls Blog – level medium
Author: levith4n

Description:
Owls are winking at you. Try to get the flag

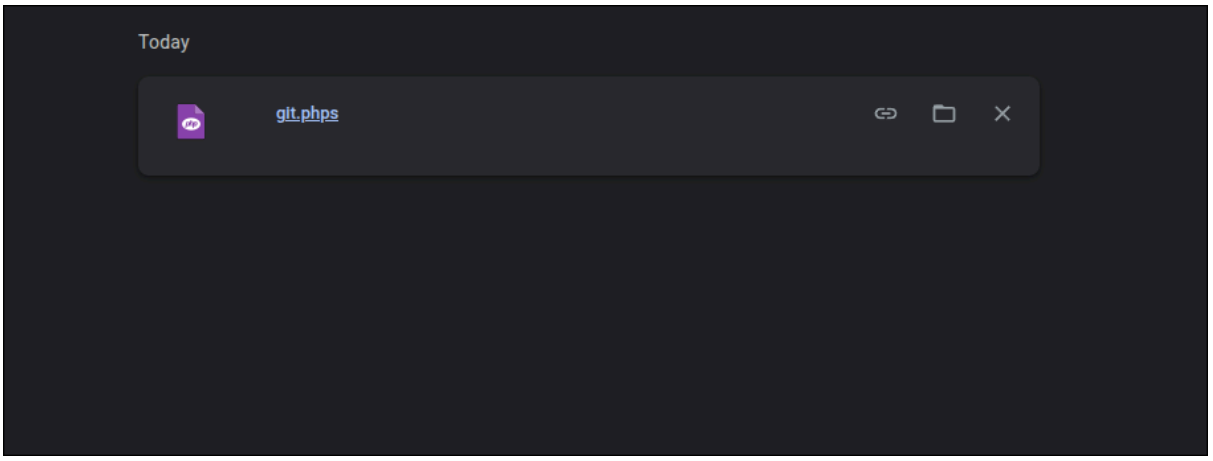
Ini adalah tampilan utama dari website target, yaitu sebuah blog.



Kemudian, saya melakukan *information gathering*, yaitu *directory bruteforcing*, pada hasilnya saya menemukan *file* **robots.txt**, file ini digunakan agar *web crawler* (seperti Googlebot) untuk mematuhi aturan yang ada di dalamnya.

Pada kasus ini, kita menemukan sebuah *file* **git.phps** tidak mengizinkan *web crawler* untuk meng-*crawling file* ini dan menampilkannya pada pencarian. Hal ini tentu mencurigakan, karena untuk apa seorang *developer* atau *sysadmin* menutupi *file* ini. File **.phps** adalah file yang diformat untuk *syntax highlighting*. Ini biasanya digunakan untuk tujuan berbagi kode di web agar lebih mudah dibaca, bukan untuk dieksekusi oleh server.

Selanjutnya, karena saya tahu bahwa *file* ini adalah sebuah *source code* PHP, saya mencoba untuk mengunjunginya untuk mendapatkan informasi berharga. Pada saat saya mengunjunginya, *file* ini otomatis terunduh.

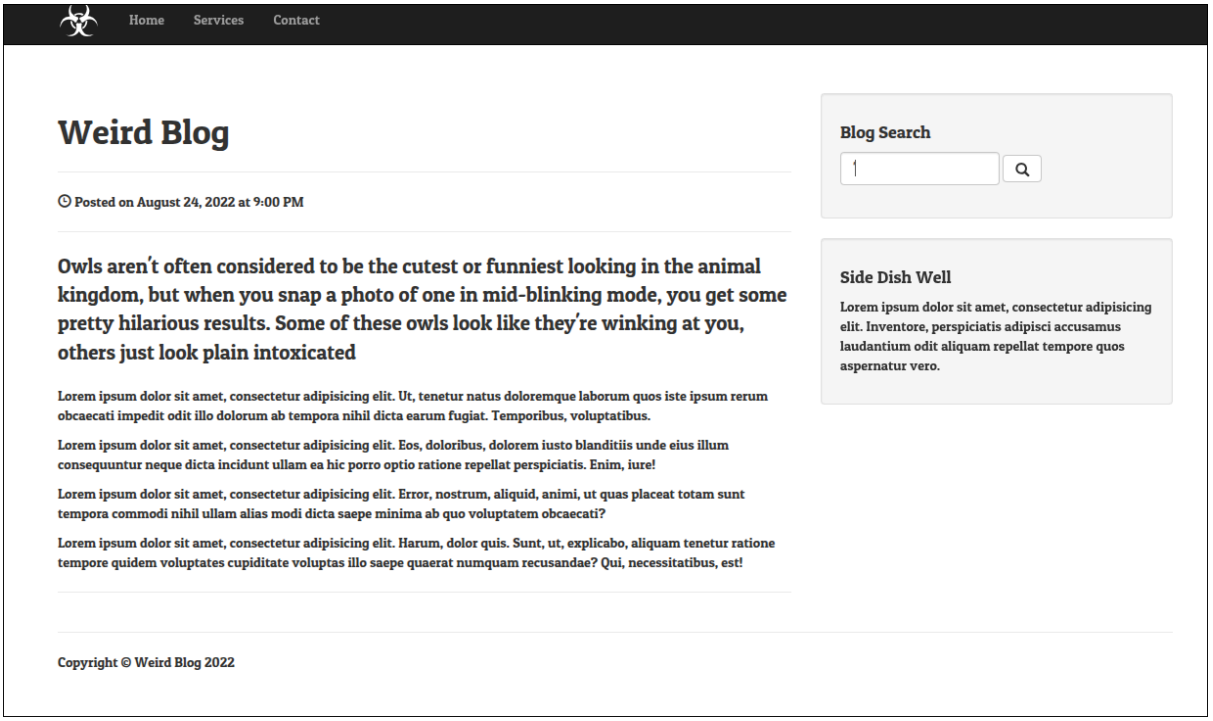


Setelah itu, saya mencoba membaca konten dari *file* tersebut. Saya menemukan sebuah kode PHP yang mengatur logika dari pencarian. Secara singkat, terdapat *regex* yang hanya mengizinkan pola pencarian sebagai berikut:

- Hanya boleh diawali **string**, **integer** dan satu karakter **"-"**.
- Tidak boleh mengandung atau diakhiri selain **string** dan **integer**.
- Modifier `m` (multiline) mengubah perilaku jangkar `^` dan `$` agar masing-masing cocok dengan **awal dan akhir dari setiap baris** (dibatasi oleh karakter newline `\n`), bukan hanya awal dan akhir dari keseluruhan string. Karena fungsi `preg_match` akan langsung berhenti dan mengembalikan `true` setelah **menemukan satu baris saja yang valid**, perilaku ini dapat kita manfaatkan untuk mem-*bypass* pola *regex* ini. Cukup dengan memastikan baris pertama cocok dengan pola regex, maka baris-baris berikutnya—bahkan yang berisi karakter tidak valid—akan diabaikan.

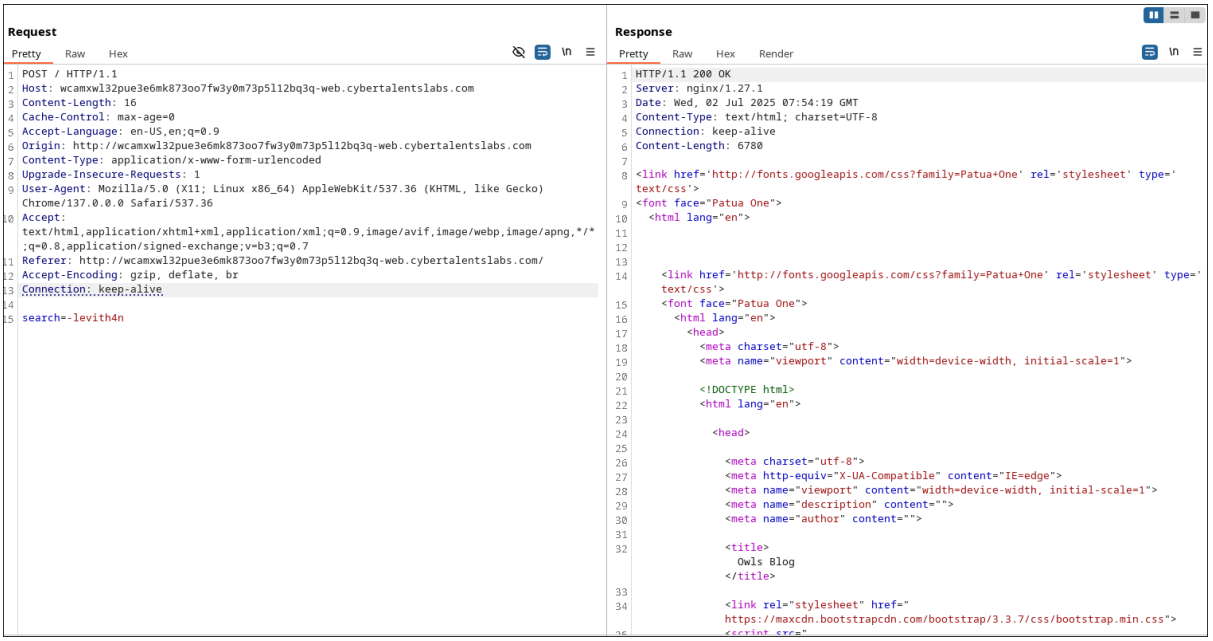
```php
<?php

$search = $POST['search'];
if (!preg_match('/^-?[0-9a-z]+$/m', $POST["search"])) {
  die("<h1><font color=\"red\">Hack Detected");
}

$query = "SELECT * FROM topics where topicname like '%$search%'";
$res = mysql_query($query);
$val = mysql_fetch_array($res);

?>
```

Saya mencoba memvalidasi *source code* tersebut, kebetulan pada tampilan utama terdapat kolom pencarian, saya mencoba memasukkan karakter ‘.

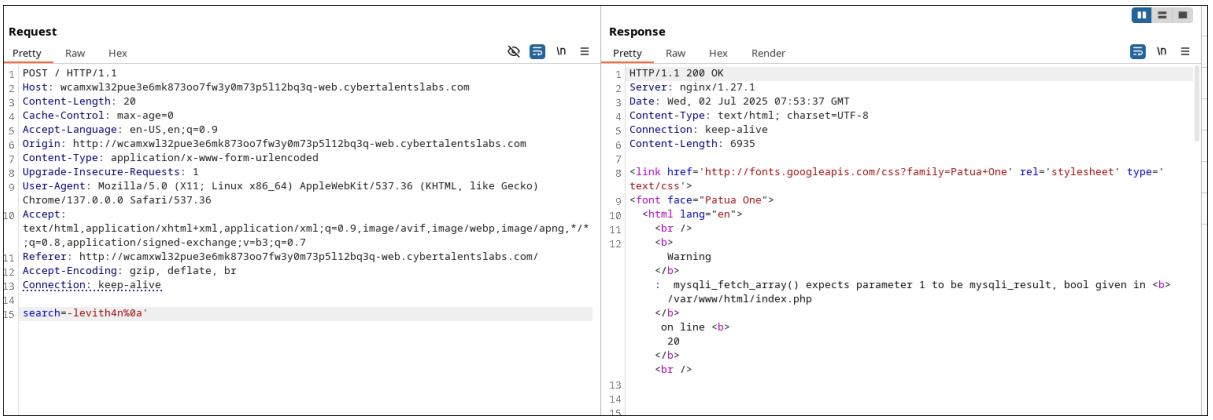


Setelah itu saya langsung diarahkan ke halaman yang terdapat tulisan “**Hack Detected**”. Hal ini mengkonfirmasi bahwa *source code* tersebut benar.

Selanjutnya, saya mencoba untuk melakukan *bypass* terhadap pola *regex* ini. Pertama-tama saya memasukkan string biasa untuk melihat respon normal.



Kemudian, saya mulai meng-*inject* menggunakan karakter *newline* \n atau **%0a** jika **url-encoding** dan diikuti karakter **‘**. Seperti yang terlihat, kita berhasil *bypass* pola *regex* ini, yaitu kita mendapatkan pesan kesalahan yang menandakan **inject** menggunakan karakter **‘** berhasil.



Selanjutnya saya melakukan **union injection** untuk menentukan jumlah kolom pada table yang terhubung dengan aplikasi web. Seperti yang terlihat jika saya masukkan dua kolom akan muncul pesan kesalahan, sehingga ini memberikan kesimpulan bahwa hanya terdapat **satu** kolom saja.

Kemudian saya mengecek versi dan DBMS apa yang digunakan menggunakan fungsi **version()**, terlihat bahwa aplikasi web menggunakan **MariaDB** versi **10.2.23** dengan sistem operasi debian.

Karena saya sudah mengetahui tipe **DBMS** yang digunakan, selanjutnya saya menggunakan fungsi **database()** untuk mengetahui nama dari database yang terhubung dengan aplikasi web. Terlihat, nama dari database yang terhubung dengan aplikasi web adalah **webctf**.

Kemudian, dengan informasi nama database sudah didapatkan, saya lanjut mencari tahu daftar table yang ada pada database tersebut, seperti yang terlihat, nama dari table yang ada pada database ini adalah **flag**.

Kemudian, saya mencoba mencari tahu kolom apa saja yang terdapat pada table ini, ternyata nama kolomnya pun juga **flag**.

Terakhir, saya hanya perlu mengekstrak nilai dari **kolom** tersebut dan mendapatkan flagnya.